TECNOLOGIA ETHERVOLTZ PARA ELEIÇÕES AUDITÁVEIS.

Matheus Faria de Alencar, Marize Correa Simoes, Diógenes Ramos da Silva

Faculdade ETEP/Faculdade de Tecnologia de São José dos Campos, Avenida Barão do Rio Branco, 882, Jardim Esplanada – 12242-800 - São José dos Campos-SP, Brasil, mtsalenc@gmail.com, marize.simoes@etep.edu.br, diogenes.silva@etep.edu.br

Resumo - Nos sistemas de votação atuais, todas as evidências geradas no momento do voto ficam sob custódia do administrador do processo eleitoral para apuração e auditorias. Os problemas deste tipo de centralização de dados e poderes ficam evidentes no Brasil, nos relatos de burlas intencionais, restrições em auditorias e punição para auditores que encontram falhas no sistema. A adoção em massa da internet permitiu o desenvolvimento de uma nova classe de programas distribuídos que substituem autoridades centrais por uma máquina virtual distribuída para garantir consenso sobre o estado de uma aplicação. Este artigo apresenta uma prova de conceito de sistema eleitoral independente de software que remove do administrador a responsabilidade de garantir a disponibilidade, integridade e confiabilidade dos registros digitais dos votos ao transformar o voto do eleitor em uma criptomoeda para descentralizar parte das provas geradas e facilitar auditorias.

Palavras-chave: tecnologia, blockchain, ethereum, criptomoeda, ethervoltz.
Área do Conhecimento: Engenharia da Computação

Introdução

Em sistemas de votação de primeira, segunda e terceira geração (BRUNAZO, 2014), todas as provas geradas pelos votos ficam sob controle do administrador e precisam ser levadas dos locais de votação para as centrais de apuração dos votos. Além dos custos envolvidos para garantir que estes equipamentos não sejam alterados ou destruídos durante o transporte, o administrador também precisa guardar estes registros após a apuração de votos para auditorias.

Após o período eleitoral, caso um cidadão queira auditar os resultados das eleições, ele precisa interagir com o administrador do processo eleitoral para ter acesso aos registros digitais, aos equipamentos e aos registros independentes de *software* (caso o administrador tenha optado por um sistema eleitoral independente de *software*). Cabe ao administrador decidir se ele tem ou não permissão para realizar a auditoria e quais são condições para a realização da mesma.

Os problemas deste tipo de centralização de poderes ficam evidentes nas eleições do Brasil. Embora não seja objetivo deste artigo discutir esses problemas, três casos são brevemente listados:

1. O Caso Marília, SP - 2004: Em auditoria, os Arquivos de Espelhos de Boletins de Urna da 400º Zona Eleitoral indicavam que muitas seções eleitorais tiveram seus resultados recebidos para apuração *antes* do início da votação (SÉRVULO et al., 2010);
2. O Caso Alagoas - 2006: Diversas irregularidades nos arquivos gerados pelas urnas foram detectadas por auditores externos. Frente as evidências, o administrador *negou* acesso aos arquivos solicitados pelos auditores e transferiu ao requerente uma cobrança antecipada no valor de R$ 2 milhões para que fosse desenvolvida uma perícia das urnas. Diante do não pagamento do valor proibitivo, o requerente foi multado e condenado por litigância de má-fé (SÉRVULO et al., 2010);
3. O Caso Itajaí, SC - 2008: Nenhuma urna preparada para a votação passou pelo teste obrigatório prescrito pelo Art. 32 da Res. TSE 22.712/08. Um caso foi o da 97ª Zona Eleitoral onde a urna da seção 236 que foi sorteada para o teste obrigatório foi substituída por outra na hora do teste, preparada exclusivamente para este fim. A urna que foi utilizada para o teste foi posteriormente colocada à parte e recarregada, procedimento que destruiu eventuais provas nela gravadas (SÉRVULO et al., 2010).

Em países como Alemanha, Argentina e Equador, que utilizam urnas mais modernas como as de segunda e terceira geração (BRUNAZO, 2014), o Princípio da Independência de Software em

Sistemas Eleitorais (RIVEST; WACK, 2006) é preservado, entretanto, todas as provas geradas no momento do voto seguem sob controle do administrador. Auditorias ocorrem apenas com a autorização e sob condições impostas pelo mesmo. O caminho do voto da urna até a apuração funciona como uma caixa preta em que o eleitor precisa confiar no administrador e em todos envolvidos no processo.

O objetivo do projeto é apresentar um modelo de sistema eleitoral que respeita o Princípio da Independência de Software em Sistemas Eleitorais e que descentralize o destino das provas geradas em cada voto, de forma que os registros físicos ficam sob custódia do administrador e os registros digitais ficam sob controle de um programa autônomo que pode ser auditado de sem a necessidade de interagir com o administrador.

Material e Métodos

Para o desenvolvimento do projeto, foram estabelecidos os seguintes requisistos:

1. Velocidade de Apuração – O sistema precisa entregar velocidade de apuração igual ou superior a dos sistemas de votação atuais;
2. Disponibilidade – Os registros digitais de votos precisam estar armazenados em um sistema confiável, que seja resistente a ataques de negação de serviço e devem estar sempre disponíveis para auditorias;
3. Integridade – Os registros digitais dos votos precisam ser imunes a alteração não autorizada e *logs* de alterações executadas devem imutáveis e permantentes;
4. Decentralizado e Autônomo – Parte do software e infraestrutura utilizados não devem estar sob controle de uma autoridade central;
5. Independência de Software – Erros ou alterações não detectados no software do sistema não devem poder causar modificações ou erros indetectáveis no resultado final.

Para garantir os requisitos 1,3 e 4, foi adotada a estratégia de transformar cada voto em uma criptomoeda (CHOHAN, 2017) batizada de VoltToken, cujas regras de emissão e transferência são definidas em um contrato inteligente programado na linguagem *Solidity*.

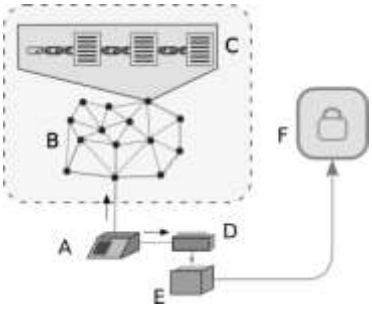
A linguagem *Solidity* é uma linguagem de alto nível similar ao javascript, orientada a contratos inteligentes e que é compilada para *bytecode* que a máquina virtual *Ethereum* é capaz de interpretar. (REITWIESSNER et al., 2017).

Ethereum é, em um sentido técnico, um computador global formado por computadores conectados e que possuem uma cópia do blockchain *Ethereum*. Este computador global possui um único processo, mas tanta memória quanto for necessária (VOGELSTELLER et al., 2017).

Alterações de estado são executados por qualquer pessoa que queira dedicar hardware e eletricidade para isso e recebem uma recompensa financeira, paga pelos usuários da rede. Essas pessoas são popularmente conhecidas com “mineradores”.

De maneira similar a outras aplicações distribuídas como o *Bitcoin* (CHOHAN, 2017), a infraestutura não possui uma autoridade central com poder de emitir ou realizar transferências de VoltTokens de forma indetectável ou fora das regras de negócio definidas no contrato inteligente.

Figura 1 – Arquitetura do sistema



Fonte: O autor.

Na Figura 1, as letras representam respectivamente:

A. Um computador a sob controle do adminstrador que o eleitor usa para votar e que possui a chave privada da carteira associada à urna, capaz de assinar transferências de VoltTokens para carteiras associadas a candidatos;
B. A rede *peer-to-peer* formada pelos milhares de nós e mineradores da rede *Ethereum.*
C. A cópia do *blockchain Ethereum* em um nó na rede;
D. Uma impressora para realizar a impressão de Votos Impressos Conferídos Pelos Eleitores (*VICE)* (BRUNAZO, 2014);
E. Uma caixa lacrada para a coleta dos *VICEs*;
F. Um cofre sob controle do adminstrador para armazenamento dos *VICE*.

Assim como nos sistemas de votação atuais, no EtherVoltz as chaves privadas que assinam cada voto estão sob controle do administrador, estão sujeitas a ataques internos e externos que podem causar o vazamento das mesmas. De maneira similar, o *frontend* utilizado para interagir com a aplicação hospedada na rede *Ethereum* pode conter erros ou estar infectado com código malicioso capaz de fraudar votos.

Para garantir que não seja possível este tipo de vulnerabilidade ser explorada para a emissão de votos fraudulentos e válidos, é necessário que o sistema atenda ao Princípio da Independência de Software em Sistemas Eleitorais, que é o item 5 dos requisitos. A estratégia utilizada é a emissão de uma versão modificada da prova auditável pelo eleitor utilizada em urnas de 2ª geração. Inclui informações adicionais que criam um laço entre ela e o registro digital do voto no *blockchain*. Uma sugestão dessa prova é apresentada na Figura 2.

Figura 2 – VICE do Ethervoltz

Fonte: O autor.

Cada transação confirmada e incluída em um bloco na rede *Ethereum*, produz um *hash* que a identifica de forma única (VOGELSTELLER et al., 2017). Uma busca com um explorador de blocos usando este *hash*, permite saber a chave pública que a assinou a transação e a chave pública da carteira destino. O EtherVoltz explora este fato para criar um laço entre as provas impressas e os registros digitais ao incluir este *hash* nos VICEs. Assim, auditando o *hash* de uma prova impressa, pode se descobrir de qual carteira de urna um determinado voto saiu e qual candidato recebeu o voto. Se a alguma informação retornada pela máquina virtual divergir do que está impresso no VICE, está configurada a fraude.

Cada voto é uma transferência de uma carteira associada a uma determinada urna, para uma carteira associada a um determinado candidato. A apuração dos votos é um procedimento instantâneo e consiste em solicitar o balanço das carteiras associadas a cada candidato.

Como o código do contrato é aberto e as transações são asseguradas por criptografia de chaves assimétricas (ETHEREUM, 2017), um auditor pode verificar as regras de negócio do sistema e verificar a origem de todas as transações realizadas.

Resultados

Um sistema de votação independente de software e que grava os registros digitais de votos numa base de dados autônoma e resistente a censura. Foi escrito um contrato inteligente na linguagem *Solidity* utilizando o *framework Truffle*. Cada função criada possui testes unitários desenvolvidos utilizando *javascript*, as suites de asserção *mocha.js* e *chai.js* e o cliente RPC *testrpc* para a execução dos testes automatizados.

O contrato inteligente define as seguintes regras, que são imutáveis após hospedagem na rede:

- A transferência de VoltTokens *para candidatos* só pode ocorrer durante o período eleitoral;
- Apenas endereços de carteiras que representam candidatos podem receber VoltTokens;
- O número total de VoltTokens em circulação é finito e sua quantidade é definida em código.
- Nenhuma nova unidade da moeda pode ser emitida após a criação do sistema;
- Cada urna recebe precisamente o número de VoltTokens correspondente ao número de eleitores que devem votar naquela urna;
- O administrador só pode definir candidatos e urnas antes do período eleitoral;
- O administrador só pode distribuir VoltTokens às urnas antes do período eleitoral.

As regras anteriores são auditáveis por qualquer pessoa através do código fonte do contrato disponibilizado pelo administrador. Todos os VoltTokens são rastreáveis desde o momento de sua emissão através de um explorador de blocos, portanto, roubos de votos ficam registrados permanentemente no blockchain assim como a origem e destino dos mesmos.

Após a execução de testes em ambiente controlado, a aplicação distribuída foi lançada na rede pública *Rinkeby* para simular uma eleição e funcionamento na internet. Foram criados pares de chaves pública e privada para o administrador, Urnas A e B e Candidatos A e B. A Tabela 1 apresenta as chaves públicas dos atores relevantes da simulação de eleição realizada.

Tabela 1 – Chaves públicas das carteiras utilizadas.

Ator	**Chave Pública**
Administrador	0x51e6dd45486b5fafeda75595b7501891c9fc54e7
Candidato A	0x2ec72e4e7846e33bd0cc88cbaecdf4bb01bcd3ff
Candidato B	0x2330d7654399d22a750bd22b8fc8501a347b7547
Urna A	0x0caa969e554a35f1176d739e384045691d21ee64
Urna B	0xdc2b8ea73104807285a3fad17c35dcc80e54ba46
Contrato Inteligente	0x063407a72493c8058b415f50076bc990c3927958

Fonte: O autor

Após a criação da aplicação na rede, o único endereço reconhecido pelo programa é o do administrador, que foi o criador do contrato. Foram enviadas transações assinadas com a chave privada do adminstrador para adicionar os endereços das urnas e dos candidatos, para que possam emitir e receber votos, respectivamente.

De maneira similar, após o lançamento da aplicação, o administrador possui custódia de todos os VoltTokens em circulação. Dos 500.000 VoltTokens criados, 20 foram transferidos para a Urna A e 5000 para a Urna B para que as urnas possam assinar transferências de suas próprias carteiras para

as carteiras dos candidatos. Dentro do período eleitoral, as duas carteiras assinaram votos para cada candidato, como ocorreria em uma eleição real.

Com o explorador de blocos *rinkeby.etherscan.io*, foram adquiridas informações das transferências de VoltTokens realizadas. 7 destas transferências estão listadas Tabela 2 em ordem cronológica, sendo que as transferências mais recentes estão no topo. As duas primeiras são a distribuição de VoltTokens realizada pelo administrador antes do período eleitoral e as demais são os votos realizados pelos eleitores dentro do período eleitoral. A coluna "Tx Hash" apresenta o *hash* de cada transação realizada e incluída em um bloco por um minerador da rede *Rinkeby*.

Tabela 2 – Transferências de VoltTokens.

Tx Hash	De	Para	Quantidade
0xe7bc3eee0198af8...	0x0caa969e554a35f...	0x2330d7654399d2...	1
0x53ff40ddba85bf4e...	0xdc2b8ea7310480...	0x2330d7654399d2...	1
0xad7336b2cefad9e...	0xdc2b8ea7310480...	0x2330d7654399d2...	1
0x4b41b2b901be65...	0xdc2b8ea7310480...	0x2330d7654399d2...	1
0x3a131d962da6916...	0xdc2b8ea7310480...	0x2330d7654399d2...	1
0x4d2c5eb13346c6...	0x51e6dd45486b5fa...	0x0caa969e554a35f...	20
0xcd463fc2767af00...	0x51e6dd45486b5fa...	0xdc2b8ea7310480...	5000

Fonte: O autor.

Discussão

Em sistemas eleitorais de primeira, segunda e terceira geração, todo o processo de auditoria precisa necessariamente envolver o administrador, já que este tem custódia de todas as provas do processo eleitoral. Já no sistema proposto, o administrador do sistema eleitoral não possui custódia nem do programa que realiza a transferência dos VoltTokens que representam os votos e nem do banco de dados que armazena os registros digitais de voto. Consequentemente, parte da auditoria pode ocorrer sem a necessidade do envolvimento do administrador. De fato, utilizando apenas um navegador de blocos, qualquer pessoa pode analisar as transferências de VoltTokens em busca de endereços e carteiras que não foram anunciados em cerimônia oficial pelo administrador.

Uma auditoria de uma urna, poderia ser conduzida da seguinte forma:

1. O auditor utiliza um explorador de blocos como o *rinkeby.etherscan.io*, para analisar todas as transações realizadas pela urna que está sendo auditada;
2. O auditor solicita ao administrador da votação, a caixa contendo os votos impressos conferíveis pelo eleitor, da mesma urna;
3. O auditor compara as duas provas, em busca de provas inconsistentes.

Alguns exemplos de inconsistências nas provas, e que caracterizam fraudes são listados:

- O número de *VICE's* na caixa entregue pelo auditor é diferente do número de registros de voto digital retornados pela máquina virtual *Ethereum*;
- Algum registro digital de voto retornado pela máquina virtual Ethereum não possui seu *VICE* associado na caixa que o administrador entregou ao auditor;
- Algum *VICE* não possui um hash válido.

Um *hash* impresso no *VICE* é considerado válido se:

- Ele existir no blockchain *Ethereum*;
- O endereço do remetente for igual ao da urna que está sendo auditada;
- O endereço da carteira do candidato for o mesmo que o impresso no *VICE* correspondente;
- O endereço do contrato da criptomoeda for o mesmo que o publicado em cerimônia oficial.

Conclusão

EtherVoltz é uma proposta que visa descentralizar o processo eleitoral e de auditorias ao transferir a responsabilidade de gerenciar dos registros digitais de votos a um programa que executa em uma máquina virtual que não possui autoridade central, é imutável e resistente a censura. Ao transformar o voto do eleitor em uma criptomoeda, o sistema imediatamente ganha todas as propriedades de segurança do protocolo de consenso e de disponibilidade da rede *peer-to-peer* nativa da plataforma. Votos são finitos e cada voto é rastreável desde a sua emissão e distribuição para as carteiras das urnas até a carteira que representa o candidato.

Embora ainda exista a necessidade da emissão e controle das provas impressas para garantir o Princípio da Independência de Software ao sistema, a separação do destino das duas provas produzidas no momento do voto dá aos eleitores poder de auditoria com limitada necessidade do envolvimento de intermediários.

Referências

BRUNAZO, A.F. Modelos e Gerações dos Equipamentos de Votação Eletrônica. Disponível em: <http://www.brunazo.eng.br/voto-e/textos/modelosUE.htm>. Acesso em: 27 jul. 2017.

CHOHAN, U.W. Criptocurrencies: A Brief Thematic Review. 2017. Disponível em: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3024330>. Acesso em: 13 ago. 2017.

REITWIESSNER, C. et al. Solidity Documentation. 2017. Disponível em: <https://solidity.readthedocs.io/en/latest/>. Acesso em: 6 jun. 2017.

RIVEST, R.L.; WACK, J.P. On the notion of “software independence“ in voting systems. Cambridge, MA. MIT. 2006. Disponível em: <https://people.csail.mit.edu/rivest/pubs/RW06.pdf>. Acesso em: 22 jul. 2017.

SÉRVULO, S.S. et al. 1º Relatório do Comitê Multidisciplinar Independente. 2010. Disponível em: <http://www.votoseguro.org/textos/CMind-1-Brasil-2010.pdf>. Acesso em: 15 jul. 2017.

VOGELSTELLER, F. et al. Ethereum White Paper. 2017. Disponível em: <https://github.com/ethereum/wiki/wiki/White-Paper>. Acesso em: 9 jul. 2017.

VOGELSTELLER, F. et al. What Is Ethereum. 2017. Disponível em: <https://github.com/ethereum/wiki/wiki/What-is-Ethereum>. Acesso em: 3 jun. 2017.